

Technische Universität Braunschweig

Institut für Programmierung und Reaktive Systeme

Programmieren I

Dr. Werner Struckmann

31. März 2008

Name:

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Vorname:

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Matrikelnummer:

--	--	--	--	--	--	--

Kennnummer:

Anrede: Frau Herr

Studiengang: Bachelor Master Diplom Magister

Fachrichtung: Informatik Wirtschaftsinformatik IST Mathematik

Mobilität u. Verkehr Physik Maschinenbau Sonstiges: _____

Die Bearbeitungszeit beträgt 120 Minuten. Die Klausur besteht aus 7 Aufgaben. Sie haben die Klausur bestanden, wenn Sie mindestens 39 von 78 möglichen Punkten erreicht haben.

Aufgabe	1	2	3	4	5	6	7	Σ
max. Punkte	8	12	8	10	6	24	10	78
Punkte								

Note:

Bitte prägen Sie sich Ihre Kennnummer gut ein. Aus Datenschutzgründen wird das Klausurergebnis nur unter dieser Kennnummer bekannt gegeben. Aus den gleichen Gründen können Ergebnisse weder telefonisch noch per E-Mail mitgeteilt werden.

Die Ergebnisse der Klausur erfahren Sie ab dem 4. April 2008 auf der WWW-Seite zu dieser Veranstaltung. Ihre Klausur können Sie am

Mittwoch, den 9. April 2008,

von 9:00 bis 11:00 Uhr und von 13:00 bis 15:00 Uhr im Raum 251 des Informatikzentrums einsehen.

Aufgabe 1: (*Darstellung von Werten*) Schreiben Sie die Dezimalzahl 92 als Oktal- und als Hexadezimalzahl. Geben Sie die Bitfolge an, durch die der Byte-Wert -23 in Java gespeichert wird. Welchen Wert gibt `System.out.println(15>>2)` aus?

8 Punkte

Lösung:

- a) 92 Oktalzahl: 134 $92 = 1 \cdot 8^2 + 3 \cdot 8 + 4$
b) 92 Hexadezimalzahl: $5C$ $92 = 5 \cdot 16 + 12$
c) Bitfolge des Byte-Werts -23 : 11101001 $-23 = -128 + 105$
d) `System.out.println(15>>2)`: 3 $15 \cong 1111, 3 \cong 0011$

Aufgabe 2: (*Wiederholungsanweisungen*) Welche Werte besitzen die Int-Variablen a und b, nachdem die folgenden Anweisungen ausgeführt wurden?

a)

```
a = 4;
b = 0;
while (a >= -4) {
    a = a - 2;
    b = 1 - 2 * b;
}
```

a = -6 b = 11

b)

```
a = -1;
b = 2;
for (int i = a+2; i <= a+4; i=2*i) {
    a = a + 2;
    b = 1 - b;
}
```

a = 7 b = 2

c)

```
a = 4;
b = 2;
do {
    a = a + b;
    b = b + 1;
} while ((a % 8) != 0);
```

a = 24 b = 7

12 Punkte

Aufgabe 3: (*Rekursion*) Gegeben sei die folgende rekursiv definierte Methode f:

```
static int f(int n, int m) {
    if ((n == 0) || (n == m) || (m == 0))
        return 1;
    else
        return f(n-1,m-1)+f(n-1,m);
}
```

Welchen Wert liefert ein Aufruf von $f(4,2)$? In welcher Reihenfolge und mit welchen Parametern wird f dabei rekursiv aufgerufen? Geben Sie die Reihenfolge der Aufrufe explizit an.

8 Punkte

Lösung:

1. Aufruf: $f(4, 2)$
2. Aufruf: $f(3, 1)$
3. Aufruf: $f(2, 0)$
4. Aufruf: $f(2, 1)$
5. Aufruf: $f(1, 0)$
6. Aufruf: $f(1, 1)$
7. Aufruf: $f(3, 2)$
8. Aufruf: $f(2, 1)$
9. Aufruf: $f(1, 0)$
10. Aufruf: $f(1, 1)$
11. Aufruf: $f(2, 2)$

$f(4, 2) = 6$

Anzahl der Aufrufe: 11

Aufgabe 4: (*Objektorientierung*) Bitte kreuzen Sie an. Für jede richtige Antwort erhalten Sie einen Punkt, für jede falsche Antwort wird ein Punkt abgezogen. Kein Kreuz bzw. zwei Kreuze bedeuten 0 Punkte. Die minimale Gesamtpunktzahl für diese Aufgabe beträgt 0 Punkte. Alle Fragen dieser Aufgabe beziehen sich auf Java.

	wahr	falsch
Eine Methode kann stets auf alle Instanzvariablen ihrer Klasse ohne Punktnotation zugreifen.		✓
<code>this</code> ist eine Variable, die auf das aktuelle Objekt zeigt.	✓	
In der Anweisung <code>return e</code> muss der Ausdruck <code>e</code> vom Rückgabetyt der zugehörigen Methode sein.		✓
Ein Konstruktor ist eine Methode ohne Rückgabewert.	✓	
Ein Konstruktor ist eine Methode mit dem Namen der zugehörigen Klasse.	✓	
In jeder Klasse kann höchstens eine Klassenvariable deklariert werden.		✓
Zwei abgeleitete Klassen können die gleiche Basisklasse besitzen.	✓	
Es gibt Variable, die während ihrer Lebensdauer auf Objekte verschiedener Typen zeigen können.	✓	
Eine Klasse, die eine <code>private</code> -Methode enthält, kann abgeleitet werden.	✓	
Eine abstrakte Klasse kann abgeleitet werden.	✓	

10 Punkte

Aufgabe 5: (*Operatoren*) Welchen Wert hat die Variable `x` vom Typ `int` nach Ausführung der folgenden Anweisung? Setzen Sie jeweils voraus, dass `x` vor der Ausführung den Wert 23 besitzt.

- a) `x = 29 / 4;` `x` besitzt den Wert 7
- b) `x = 29 % 4;` `x` besitzt den Wert 1
- c) `x += 29 - 4;` `x` besitzt den Wert 48
- d) `x -= 29 + 4;` `x` besitzt den Wert -10
- e) `x *= 29 / 4;` `x` besitzt den Wert 161
- f) `x /= 29 % 4;` `x` besitzt den Wert 23

6 Punkte

Aufgabe 6: (*Programmerstellung*) Schreiben Sie eine Methode

```
static boolean sameElements(int[] a, int[] b),
```

die als Parameter zwei Felder *a* und *b* ganzer Zahlen erhält. Die Methode soll den Wert `true` liefern, falls beide Felder die gleichen Elemente enthalten. Im anderen Fall soll das Ergebnis `false` sein. Beachten Sie, dass die Parameterfelder *a* und *b* leer sein können. Falls ein Null-Zeiger als Parameter übergeben wird, soll eine Ausnahme ausgelöst werden.

Beispielsweise enthalten die Felder *r*: 1,1,1,3,4,2,3,5 und *s*: 1,2,3,4,5 die gleichen Elemente. Für diese Felder soll der Aufruf `sameElements(int[] r, int[] s)` daher `true` ergeben.

24 Punkte

Lösung:

```
static boolean contains(int[] a, int[] b) {
  l: for (int x : a) {
    for (int y : b)
      if (x == y) continue l;
    return false;
  }
  return true;
}

static boolean sameElements(int[] a, int[] b) {
  assert a != null && b != null;
  return contains(a,b) && contains(b,a);
}
```

Aufgabe 7: (*Programmzuverlässigkeit*) Gegeben sei die folgende Java-Methode:

```
static int f(int n) {
    assert n >= 1;          // Vorbedingung P
    int i = 1,
        s = 0;
    while (i <= n) {
        s += i*(3*i-1);
        i++;
    }
    assert ...;           // Nachbedingung Q (*)
    return s;
}
```

- Welche Werte besitzen die Variablen i und s zum Zeitpunkt der Ausführung der `assert`-Anweisung (*)? Formulieren Sie eine entsprechende Nachbedingung Q .
- Schreiben Sie Q als Java-Ausdruck, sodass dieser in der `assert`-Anweisung (*) verwendet werden kann.
- Wie lautet eine zum Nachweis der partiellen Korrektheit bezüglich P und Q geeignete Schleifeninvariante?

Zur Erinnerung: Es ist $\sum_{j=1}^n j^2 = \frac{1}{6}n(n+1)(2n+1)$.

10 Punkte

Lösung: Für die Variablen i und s gelten $i = n + 1$ und $s = \sum_{j=1}^n j(3j - 1)$. Daher ist `i == n+1 && s == n*n*(n+1)` eine mögliche Nachbedingung Q . Eine zum Nachweis der partiellen Korrektheit geeignete Schleifeninvariante ist $s = \sum_{j=1}^{i-1} j(3j - 1) \wedge 1 \leq i \leq n + 1$ bzw. $s = i(i - 1)^2 \wedge 1 \leq i \leq n + 1$.